

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 09-160759
 (43)Date of publication of application : 20.06.1997

(51)Int.Cl. G06F 9/06
 G06F 15/163

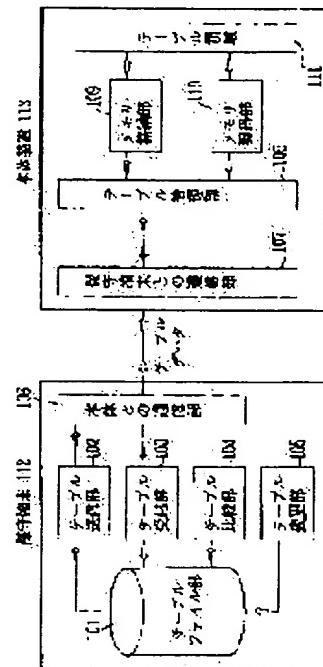
(21)Application number : 07-318004 (71)Applicant : FUJITSU LTD
 (22)Date of filing : 06.12.1995 (72)Inventor : SUZUKI RITSU

(54) MANAGEMENT SYSTEM FOR OPERATION DEFINITION INFORMATION TABLE OF COMPUTER SYSTEM

(57)Abstract:

PROBLEM TO BE SOLVED: To safely change the table contents of a main body device, to shorten the operation stop time of the main body device and to improve the development efficiency of the table changing function in a computer system which has an operation definition information table contained in a memory of the main body device and can be connected to a maintenance terminal.

SOLUTION: A table changing part 105 of a maintenance terminal 112 changes the table file included in a table file part 101. The table contents shown by the changed table file are transferred to a main body device 113 via a table transmission part 102. A table management part 108 of the device 113 writes the table contents received from the terminal 112 into a table area 111. Thus, the table contents are managed as a table file of the terminal 112, so that a table changing task, etc., can be carried out at a place (a business office, etc.) where no device 113 is installed.



LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's

(19)日本国特許庁 (JP)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開平9-160759

(43)公開日 平成9年(1997)6月20日

(51)Int.Cl. ⁶	識別記号	序内整理番号	F I	技術表示箇所
G 0 6 F 9/06 15/163	4 1 0		G 0 6 F 9/06 15/16	4 1 0 B 3 2 0 D

審査請求 未請求 請求項の数13 ○L (全27頁)

(21)出願番号	特願平7-318004	(71)出願人	000005223 富士通株式会社 神奈川県川崎市中原区上小田中4丁目1番 1号
(22)出願日	平成7年(1995)12月6日	(72)発明者	鈴木 立 神奈川県川崎市中原区上小田中1015番地 富士通株式会社内
		(74)代理人	弁理士 大曾 義之 (外1名)

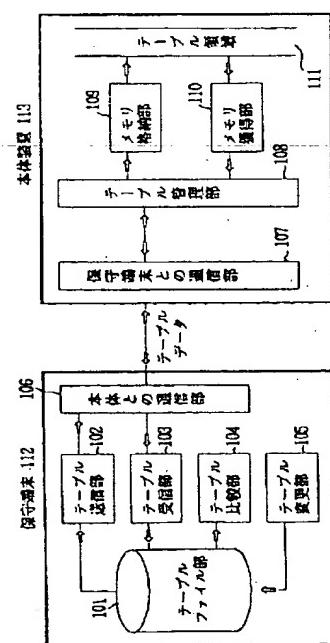
(54)【発明の名称】 コンピュータシステムの動作定義情報テーブルの管理方式

(57)【要約】

【課題】 動作定義情報テーブルを本体装置のメモリ上有し、保守端末を接続可能なコンピュータシステムに關し、本体装置上のテーブル内容の変更を安全に実施可能とすると共に、本体装置の運用停止時間を短縮し、また、テーブル変更機能の開発効率を向上させることを課題とする。

【解決手段】 保守端末112内のテーブル変更部105は、テーブルファイル部101内のテーブルファイルに対して変更動作を実行する。変更後のテーブルファイルが示すテーブル内容は、テーブル送信部102によって本体装置113に転送される。本体装置113内のテーブル管理部108は、保守端末112から転送されたテーブル内容をテーブル領域111に書き込む。このように、テーブル内容が保守端末112上のテーブルファイルとして管理されることにより、本体装置がない場所(事務所等)でのテーブル変更作業等が可能となる。

第1の実施の形態の構成図



【特許請求の範囲】

【請求項 1】 動作定義情報テーブルを本体装置のメモリ上有し、保守端末を接続可能なコンピュータシステムにおいて、
前記保守端末内に、
前記本体装置のための動作定義情報テーブルをテーブルファイルとして管理するテーブルファイル管理手段と、該テーブルファイル管理手段が管理するテーブルファイルが示すテーブル内容を、前記本体装置に送信するテーブル送信手段と、

を含み、
前記本体装置内に、
前記保守端末内のテーブル送信手段から前記テーブル内容を受信し、該テーブル内容により該本体装置が保持しているテーブル内容を置き換えるテーブル更新手段を、
含む、

ことを特徴とする動作定義情報テーブルの管理方式。

【請求項 2】 前記本体装置は、該本体装置が保持しているテーブル内容を前記保守端末に送信するテーブル送信手段を更に含み、
前記テーブルファイル管理手段は、前記テーブル送信手段により送信されたテーブル内容を前記テーブルファイルとして受信するテーブル受信手段を含む、

ことを特徴とする請求項 1 に記載の動作定義情報テーブルの管理方式。

【請求項 3】 前記テーブルファイル管理手段は、該テーブルファイル管理手段が保持するテーブルファイルの内容を、該テーブルファイル手段内のテーブル受信手段が受信したテーブルファイルの内容と比較するテーブル比較手段を含む、
ことを特徴とする請求項 1 に記載の動作定義情報テーブルの管理方式。

【請求項 4】 前記テーブルファイル管理手段は、前記テーブルファイルの内容を変更するテーブル変更手段を含む、
ことを特徴とする請求項 1 に記載の動作定義情報テーブルの管理方式。

【請求項 5】 前記テーブル変更手段は、画面の表示形式を定義する画面定義ファイルの内容と前記動作定義情報テーブルの構造を定義する属性定義ファイルの内容に基づいて、前記テーブルファイルの内容を変更する、
ことを特徴とする請求項 4 に記載の動作定義情報テーブルの管理方式。

【請求項 6】 前記動作定義情報テーブルを生成するためのテーブルソース情報及び前記動作定義情報テーブルを前記本体装置に格納するためのロケートマップ情報に基づいて、前記画面定義ファイル及び前記属性定義ファイルを生成する定義ファイル生成手段を更に有する、
ことを特徴とする請求項 5 に記載の動作定義情報テーブルの管理方式。

【請求項 7】 前記本体装置は、該本体装置が保持しているテーブル内容又は装置状態を前記保守端末に送信するテーブル送信手段を更に含み、
前記テーブルファイル管理手段は、前記テーブル送信手段により送信されたテーブル内容又は装置状態を前記テーブルファイルとして受信するテーブル受信手段を含み、

前記テーブルファイル管理手段内のテーブル変更手段は、前記テーブルファイルをリードオンリーモードでオープンする機能を有し、
前記テーブルファイル管理手段内のテーブル受信手段と前記リードオンリーモードで動作する前記テーブル変更手段が連携することにより、前記本体装置のテーブル内容又は装置状態を表示する、
ことを特徴とする請求項 4 乃至 6 の何れか 1 項に記載の動作定義情報テーブルの管理方式。

【請求項 8】 前記本体装置内に、
前記テーブル更新手段が更新する動作定義情報テーブルの更新履歴を保持するテーブル更新履歴保持手段と、
該テーブル更新履歴保持手段が保持する前記動作定義情報テーブルの更新履歴を前記保守端末に送信するテーブル更新履歴送信手段と、

を含み、
前記保守端末は、前記本体装置内のテーブル更新履歴送信手段が送信した前記動作定義情報テーブルの更新履歴を受信して表示するテーブル更新履歴表示手段を含む、
ことを特徴とする請求項 1 乃至 7 の何れか 1 項に記載の動作定義情報テーブルの管理方式。

【請求項 9】 前記本体装置内のテーブル更新手段は、
前記保守端末から前記テーブル内容を受信しそれを保持するテーブル内容保持手段と、
前記本体装置のリセット時に、前記テーブル内容保持手段が保持するテーブル内容を変換し、該変換されたテーブル内容により該本体装置が保持しているテーブル内容を置き換えるテーブル変換手段と、
を含むことを特徴とする請求項 1 乃至 8 の何れか 1 項に記載の動作定義情報テーブルの管理方式。

【請求項 10】 保守端末を接続可能で、動作定義情報テーブルの内容に基づいて動作し、共有メモリを有するマルチプロセッサシステムにおいて、
前記保守端末内に、

前記本体装置のための動作定義情報テーブルをテーブルファイルとして管理するテーブルファイル管理手段と、該テーブルファイル管理手段が管理するテーブルファイルが示すテーブル内容を、前記マルチプロセッサシステムを構成する所定のプロセッサに送信するテーブル送信手段と、
を含み、
前記所定のプロセッサ内に、

前記保守端末内のテーブル送信手段から前記テーブル内

容を受信し、該テーブル内容を前記マルチプロセッサシステムを構成する各プロセッサに転送するテーブル転送手段を、

含み、

前記各プロセッサ内に、
前記所定のプロセッサ内のテーブル転送手段から前記テーブル内容を受信し、該各プロセッサがローカルに保持しているテーブル内容を置き換えるテーブル更新手段を、

含む、

ことを特徴とする動作定義情報テーブルの管理方式。

【請求項11】 請求項1乃至10の何れか1項に記載のシステムに使用される保守端末。

【請求項12】 請求項1乃至9の何れか1項に記載のシステムに使用される本体装置。

【請求項13】 請求項10に記載のマルチプロセッサシステムを構成するプロセッサ。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】 本発明は、動作定義情報テーブルを本体装置のメモリ上に有し、保守端末を接続可能なコンピュータシステムに関する。

【0002】

【従来の技術及び発明が解決しようとする課題】 近年のコンピュータシステム、特に社会システムとして使用されるオンラインリアルタイムシステムでは、長期間の連続運転が要求されているため、機械ディスク装置等の駆動部を有する装置を使用できない。このため、装置の動作定義情報がテーブルとして、メモリ等の駆動部を持たない記憶媒体上に保存される必要がある。

【0003】 この場合に、動作定義情報のテーブルをどのように安全に更新するかが課題となる。図23は、従来技術の説明図である。

【0004】 本体装置2308上のテーブル領域2306のテーブル内容を変更するためには、オペレータは、本体装置2308と保守端末2307とを接続し、保守端末2307上のテーブル変更部2301を起動する。テーブル変更部2301でのテーブル変更機能によって変更されたテーブル内容は、本体との通信部2302を介して即座に本体装置2308に転送される。

【0005】 本体装置2308では、保守端末との通信部2303を介して受信された上記テーブル内容は、メモリ格納部2304を介してテーブル領域2306に格納される。

【0006】 また、保守端末2307内のテーブル変更部2301は現状のテーブルを表示する機能も有するが、これは、本体装置2308内のメモリ獲得部2305が、テーブル領域2306内のテーブル内容を獲得し、それを保守端末2307に送信することにより実現されていた。

【0007】 しかし、上述の従来技術では、保守端末2307を本体装置2308に直接接続しないとテーブル内容の変更を行うことができず、オペレータはテーブル内容の事前準備ができず、テーブル変更作業の安全性が阻害されるという問題点を有していた。なお、テーブル領域2306に格納されるテーブルデータは、重要性及び守秘性の高いものが多く、また、できる限り単時間での作業も要求されるため、リモートメンテナンスも困難であった。

【0008】 また、上述の従来技術では、保守端末2307内のテーブル変換部2301の機能は、プログラムロジックとして実現していた。これに対して、テーブル変更機能の画面イメージやテーブル構造は、顧客の要望等に基づいて頻繁に修正することがあるため、テーブル変換部2301の機能の開発効率・修正効率が悪いという問題点を有していた。

【0009】 更に、保守端末2307に実装されるテーブル変換部2301の機能は、本体装置2308内のテーブル領域2306に格納されるテーブルデータのロケートアドレス等と関連づけて作成する必要がある。従つて、従来は、プログラマが、本体装置2308に対応するソースプログラム及びロケートマップファイル等を参照しながら、テーブル変換部2301の機能を開発しなければならなかつた。このため、上記機能の開発効率が悪く、また、上記機能と本体装置のテーブル構造の間の認識違いに基づくバグも多発していた。

【0010】 また、従来、本体装置2308のテーブル領域2306内のテーブル内容や装置状態を表示する機能も、新たなプログラムロジックとして作成しなければならなかつたため、開発効率が悪かつた。

【0011】 更に、従来、本体装置2308のテーブル領域2306に格納されているテーブルがどうのように変更されたかを示す情報は、オペレータ自身が管理しなければならず、管理ミスが多発していた。

【0012】 また、従来、本体装置2308のテーブル領域2306内のテーブル内容を変更するためには、図24に示されるように、本体装置2308の状態を装置運用状態から装置運用停止状態に変更した上でテーブル内容の変更を実行し、その後に再び本体装置113の状態を装置運用停止状態から装置運用状態に戻す必要があった。このため、本体装置113にメンテナンスマードを実装しなければならず、また、テーブル変更が長時間に渡る場合には本体装置2308も長時間にわたってその運用が停止してしまい装置稼働率の低下を招くという問題点を有していた。

【0013】 上記従来技術とは別に、共有メモリを有する密結合マルチプロセッサシステムにおける従来のテーブル配置技術について、図25を用いて説明する。マルチプロセッサシステムにおいては、コストや操作の単純化のために、保守端末2505は、1つのプロセッサ

(システム管理CPU2506)のみに接続するのが望ましい。

【0014】従来技術では、他プロセッサ(個別機能CPU2507)において必要なテーブルは、共有メモリ2508上のテーブル領域2504に配置されていた。オンラインリアルタイムシステム等では、アプリケーション2503は、運用中に頻繁にテーブル内容を参照しなければならないものが多い。ところが、一般的に共有メモリ2508へのアクセスは、各プロセッサ内のローカルメモリへのアクセスよりも多くプロセッサ実行時間を占有するため、各プロセッサの負荷が重くなってしまい、結果としてプロセッサの数が増え、システム全体のコストアップを招いていた。

【0015】また、共有メモリ2508に対するアクセス負荷も増大するため、他プロセッサからの共有メモリ2508へのアクセスを阻害する結果も招き、システムのレスポンスを低下させてしまうという問題点を有していた。

【0016】本発明の課題は、本体装置上のテーブル内容の変更を安全に実施可能と共に、本体装置の運用停止時間を短縮し、また、テーブル変更機能の開発効率を向上させることにある。

【0017】

【課題を解決するための手段】本発明の第1の態様は、動作定義情報テーブルを本体装置のメモリ上に有し、保守端末を接続可能なコンピュータシステムを前提とする。

【0018】そして、保守端末内に、本体装置のための動作定義情報テーブルをテーブルファイルとして管理するテーブルファイル管理手段と、そのテーブルファイル管理手段が管理するテーブルファイルが示すテーブル内容を、本体装置に送信するテーブル送信手段とを有する。

【0019】また、本体装置内に、保守端末内のテーブル送信手段からテーブル内容を受信し、そのテーブル内容によりその本体装置が保持しているテーブル内容を置き換えるテーブル更新手段を有する。

【0020】以上の構成を有する本発明の第1の態様では、テーブル内容が保守端末上のテーブルファイルとして管理されることにより、本体装置がない場所(事務所等)でのテーブル変更作業等が可能となり、オペレータは事前にテーブル内容を準備することが可能となり、テーブル変更の安全性が向上する。

【0021】上述の本発明の第1の態様の構成において、本体装置は、その本体装置が保持しているテーブル内容を保守端末に送信するテーブル送信手段を更に含み、保守端末内のテーブルファイル管理手段は、テーブル送信手段により送信されたテーブル内容をテーブルファイルとして受信するテーブル受信手段を含むように構成することができる。また、テーブルファイル管理手段

は、そのテーブルファイル管理手段が保持するテーブルファイルの内容を、そのテーブルファイル手段内のテーブル受信手段が受信したテーブルファイルの内容と比較するテーブル比較手段を含むように構成することができる。また、テーブルファイル管理手段は、テーブルファイルの内容を変更するテーブル変更手段を含むように構成することができる。

【0022】更に、テーブル変更手段は、画面の表示形式を定義する画面定義ファイルの内容と動作定義情報テーブルの構造を定義する属性定義ファイルの内容に基づいて、テーブルファイルの内容を変更するように構成することができる。

【0023】このような定義ファイルを有する構成により、システム開発者は、画面イメージとテーブル構造の定義ファイルを作成するだけで、テーブル変更機能を簡単に提供することが可能となり、開発効率が向上する。

【0024】次に、テーブル変更手段が画面定義ファイル及び属性定義ファイルを有する上述の構成において、動作定義情報テーブルを生成するためのテーブルソース情報及び動作定義情報テーブルを本体装置に格納するためのロケートマップ情報に基づいて、画面定義ファイル及び属性定義ファイルを生成する定義ファイル生成手段を更に有するように構成することができる。

【0025】このような定義ファイル生成手段を有する構成により、システム開発者はほとんど作業なしでテーブル変更機能を提供でき、開発効率を向上させることができる。また、本体装置のテーブル構造と保守端末のテーブル変更手段の機能の間で、テーブル構造の認識違いに基づくバグの発生がなくなるため、テーブル変更手段の機能の信頼性を向上させることができる。

【0026】次に、ここまで本発明の構成において、本体装置は、その本体装置が保持しているテーブル内容又は装置状態を保守端末に送信するテーブル送信手段を更に含み、保守端末内のテーブルファイル管理手段は、テーブル送信手段により送信されたテーブル内容又は装置状態をテーブルファイルとして受信するテーブル受信手段を含み、テーブルファイル管理手段内のテーブル変更手段は、テーブルファイルをリードオンリーモードでオープンする機能を有し、テーブルファイル管理手段内のテーブル受信手段とリードオンリーモードで動作するテーブル変更手段が連携することにより、本体装置のテーブル内容又は装置状態を表示するように構成することができる。

【0027】このようなテーブル受信手段とリードオンリーモードを有するテーブル変更手段を連携動作させるだけで、本体装置のテーブル内容や装置状態の表示機能を簡単に実現することができる。

【0028】続いて、ここまで本発明の構成において、本体装置内に、テーブル更新手段が更新する動作定義情報テーブルの更新履歴を保持するテーブル更新履歴

保持手段と、そのテーブル更新履歴保持手段が保持する動作定義情報テーブルの更新履歴を保守端末に送信するテーブル更新履歴送信手段とを含み、保守端末は、本体装置内のテーブル更新履歴送信手段が送信した動作定義情報テーブルの更新履歴を受信して表示するテーブル更新履歴表示手段を含むように構成することができる。

【0029】このような更新履歴を管理可能な構成により、テーブル版数管理の安全性を向上させることができる。次に、ここまで本発明の構成において、本体装置内のテーブル更新手段は、保守端末からテーブル内容を受信しそれを保持するテーブル内容保持手段と、本体装置のリセット時に、テーブル内容保持手段が保持するテーブル内容を変換し、その変換されたテーブル内容によりその本体装置が保持しているテーブル内容を置き換えるテーブル変換手段とを含むように構成することができる。

【0030】このようなテーブル変換手段を含む構成により、変更されたテーブル内容が保守端末から本体装置に転送される際に本体装置の運用を停止させる必要がなく、転送終了後に本体装置をリセットするだけで本体装置のテーブル内容を更新することができるため、本体装置にメンテナンスマード等の保守を行うモードを実装する必要がなく、また、本体装置の稼働率の低下も防ぐことができる。

【0031】次に、本発明の第2の態様は、保守端末を接続可能で、動作定義情報テーブルの内容に基づいて動作し、共有メモリを有するマルチプロセッサシステムを前提とする。

【0032】そして、保守端末内に、本体装置のための動作定義情報テーブルをテーブルファイルとして管理するテーブルファイル管理手段と、そのテーブルファイル管理手段が管理するテーブルファイルが示すテーブル内容を、マルチプロセッサシステムを構成する所定のプロセッサに送信するテーブル送信手段とを含む。

【0033】また、上記所定のプロセッサ内に、保守端末内のテーブル送信手段からテーブル内容を受信し、そのテーブル内容をマルチプロセッサシステムを構成する各プロセッサに転送するテーブル転送手段を含む。

【0034】そして、上記各プロセッサ内に、所定のプロセッサ内のテーブル転送手段からテーブル内容を受信し、その各プロセッサがローカルに保持しているテーブル内容を置き換えるテーブル更新手段を含む。

【0035】上述の本発明の第2の態様の構成により、テーブル内容が各プロセッサにローカルに保持されるため、テーブル内容をアクセスするプロセッサの実行時間を短縮することができる。また、共有メモリへのアクセスの集中を回避することができ、システム全体のレスポンスを向上させることができる。

【0036】

【発明の実施の形態】以下、図面を参照しながら本発明

の実施の形態について詳細に説明する。

第1の実施の形態

図1は、本発明の第1の実施の形態の構成図であり、図2は、その動作フローチャートである。

【0037】第1の実施の形態では、本体装置113側のテーブル領域111の内容を、保守端末112側のテーブルファイル部101にテーブルファイルとして保有する点が特徴である。

【0038】オペレータは、保守端末112を使用して

10 本体装置113のテーブル領域111の内容を変更する際には、まず予め、保守端末112のテーブル受信部103を起動することにより、本体装置113内のテーブル領域111から保守端末112にテーブル内容を転送させる(図2のステップ201)。具体的には、本体装置113内のテーブル管理部108が、メモリ獲得部110を起動することによって、テーブル領域111のテーブル内容が読み出される。このテーブル内容は、本体装置113内の保守端末との通信部107と保守端末112内の本体との通信部106を介して、テーブル受信部103に転送される。テーブル受信部103は、そのテーブル内容を、テーブルファイル部101にテーブルファイルとして格納する。

【0039】次に、オペレータは、テーブル比較部104を起動することにより、テーブルファイル部101に格納されている保守端末112上の新テーブルファイルと、本体装置113から転送された旧テーブルファイルとを比較する(図2のステップ202)。

【0040】この結果、オペレータは、テーブル変更処理を継続する場合には、テーブル変更部105を起動することにより、テーブルファイル部101に格納されている新テーブルファイルの内容を変更する(図2のステップ203)。その後、再びテーブル比較部104が起動されることにより、変更された新テーブルファイルと、旧テーブルファイルとが比較される(図2のステップ203→202)。オペレータは、テーブル変更処理を継続する場合には、再びテーブル変更部105を起動する。

【0041】オペレータは、以上のテーブル変更処理を繰り返し、テーブル変更処理を終了すると、テーブル送信部102が起動され、テーブルファイル部101内の新テーブルファイルの内容が本体装置113に転送される(図2のステップ202→204)。具体的には、テーブル送信部102がテーブルファイル部101内の新テーブルファイルを読み出し、それを、保守端末112内の保守端末との通信部107及び本体装置113内の本体との通信部106を介して、本体装置113内のテーブル管理部108に転送する。テーブル管理部108は、メモリ格納部109を起動することにより、新テーブルファイルの内容をテーブル領域111に格納する。

【0042】以上の第1の実施の形態により、保守端末

112さえあればテーブルファイルを変更することができるため、オペレータは、本体装置113がない場所（事務所等）で、事前にテーブルファイルを準備することが可能となり、テーブル変更の安全性が向上する。

第2の実施の形態

図3は、本発明の第2の実施の形態の構成図である。

【0043】第2の実施の形態は、図1の第1の実施の形態におけるテーブル変更部105を実現する形態である。第2の実施の形態では、テーブルファイル部304内のテーブルファイルに対するテーブル変更に必要な画面イメージとテーブル構造が、画面定義ファイル302と属性定義ファイル303にファイル化されることにより、テーブル変更プログラム301の本体と分離される。この結果、テーブル変更部の開発・修正を行うことが容易になる。

【0044】まず、画面定義ファイル302は、テーブル変更に必要な画面イメージを定義するファイルである。このファイルの内容例は、図3の305として示される。305において、文字列“\$\$\$\$”は、オペレータが入力すべき項目（テーブル構造における項目）を示す。

【0045】次に、属性定義ファイル303は、オペレータが入力すべきテーブル構造の項目を定義するファイルである。このファイルの内容例は、図3の306として示される。306において、“\$1”等は、画面定義ファイル302内の“\$\$\$\$”（305を参照）に対応する。即ち、画面定義ファイル302内の最初の“\$\$\$\$”が属性定義ファイル303内の“\$1”に、画面定義ファイル302内の次の“\$\$\$\$”が属性定義ファイル303内の“\$2”に、それぞれ対応する。また、306において、入力項目毎に、該入力項目の本体装置上のメモリアドレス、変換形式（10進／16進形式等）、画面上での表示色、データ形式（1バイト／2バイト等）が定義される。

【0046】これらの情報をもとに、テーブル変更プログラム301が実行されると、図3の307に示されるようなテーブル変更画面が特には図示しない表示装置に表示され、テーブル変更が可能となる。画面中の反転部分がオペレータによる入力項目となる。

【0047】図4及び図5は、テーブル変更プログラム301の動作フローチャートを示す図である。テーブル変更の実行が開始されると、始めに、画面定義ファイル302の内容（図3の305を参照）が、テーブル変更プログラム301の画面領域に読み込まれる（図4のステップ401）。

【0048】次に、属性定義ファイル303の内容（図3の306を参照）が、テーブル変更プログラム301の属性領域に読み込まれる（図4のステップ402）。次に、テーブルファイル部304から、変更対象のテーブルファイルの内容が、テーブル変更プログラム301

10
のデータ領域に読み込まれる（図4のステップ403）。

【0049】続いて、上記画面領域及び属性領域の内容に従って、データ領域に読み込まれたテーブルファイルの内容（図3の307を参照）が表示される（図4のステップ404）。図5は、この図4のステップ404の詳細を示す動作フローチャートである。

【0050】まず、画面領域から読み出されるデータのうち、“\$”以外の文字が表示される（図5のステップ10 501）。次に、属性領域から属性情報が1要素づつ取り出される（図5のステップ502）。

【0051】続いて、属性情報であるアドレスに対応するデータが、データ領域から獲得される（図5のステップ503）。更に、画面領域の“\$”の部分に、上記獲得されたデータが、属性領域の属性情報に従って表示される（図5のステップ504）。

【0052】上述の図5のステップ502～504の処理が、属性領域の最後の属性情報が取り出されるまで繰り返される（図5のステップ505→502）。以上のようにして、図4のステップ404でテーブル変更画面が表示された後に、図4のステップ405～410の処理が繰り返されることにより、オペレータが操作する各種キーに対応する処理が実行される。

【0053】即ち、キー入力待ちの状態（図4のステップ405）において、オペレータがカーソルキーを押すと、ステップ406の判定がYESとなって、カーソルの移動処理が実行される（図4のステップ407）。

【0054】また、キー入力待ちの状態（図4のステップ405）において、オペレータがデータ入力をを行う30 30と、ステップ408の判定がYESとなって、データ領域の内容の変更処理が実行される（図4のステップ409）。

【0055】キー入力待ちの状態（図4のステップ405）において、オペレータが終了キーを押すと、ステップ410の判定がYESとなって、データ領域の内容がテーブルファイル部304内の該当テーブルファイルに書き込まれて（図4のステップ411）、テーブルの変更処理を終了する。

【0056】以上説明した第2の実施の形態により、システムの開発者は、テーブル変更に必要な画面定義ファイル302及び属性定義ファイル303を作成するだけで、テーブル変更時等において、各本体装置に対応したテーブル表示を行なうことが可能となるため、テーブル変更プログラム301の開発効率を向上させることが可能となる。

第3の実施の形態

図6は、本発明の第3の実施の形態の構成図である。

【0057】第3の実施の形態では、図3の第2の実施の形態におけるテーブル変更部の機能が実現される場合50 50に、画面定義ファイル及び属性定義ファイルを、本体装

11

置にロードされるテーブルに対応するロードモジュールファイルと、その元となるプログラムソースファイルから自動生成することができる。

【0058】図6の構成は、本体装置にロードされるテーブルを開発する開発装置上に実現される。図6において、まず、テーブルソースファイル602は、本体装置のテーブル領域を定義するソースファイルであり、本体装置のプログラムの一部となる。このファイルの内容例は、図6の604として示され、ここには、テーブル領域の構造が所定のプログラム言語によって記述される。

【0059】次に、ロケートマップファイル603は、本体装置にテーブル領域のプログラムがロケートされる際に、一般的に知られているロケータ機能によって自動生成されるファイルであり、テーブルソースファイル602内の各要素のメモリアドレス等を含む。

【0060】これらの情報をもとに、テーブル変更用ファイル自動生成プログラム601が実行されることにより、自動的にテーブル変更に必要な画面定義ファイル605及び属性定義ファイル606が生成される。これらは、それぞれ、第2の実施の形態における図3の画面定義ファイル302及び属性定義ファイル303に対応する。

【0061】図7は、図6のテーブル変更用ファイル自動生成プログラム601の動作フローチャートである。まず、ロケートマップファイル603の内容が、上記プログラム内のロケートマップ領域に読み込まれる（ステップ701）。

【0062】次に、テーブルソースファイル602の内容が上記プログラム内のワークデータ領域に1行ずつ読み込まれながら（ステップ702）、そのファイルの最後が検出されるまで（ステップ703の判定がYESとなるまで）、以下のステップ704～709の処理が繰り返し実行される。

【0063】即ち、まず、現在行にテーブル要素であるシンボルが存在するか否かが判定される（ステップ704）。現在行にシンボルがなく、ステップ704の判定がNOであれば、ステップ705～708は実行されず、ステップ709で、現在行の内容が単純に画面定義ファイル605に書き込まれる。

【0064】現在行にシンボル（テーブル要素）があり、ステップ704の判定がYESであれば、以下のステップ705から709が実行される。即ちまず、ワークデータ領域上の現在行内のシンボルの右側に、そのデータ型（図6の604を参照）に合わせて、文字列“\$”が追加される（ステップ705）。

【0065】次に、上記シンボル名（テーブル要素名）がロケートマップ領域から検索されて、そのアドレスが獲得される（ステップ706）。続いて、上記シンボル（テーブル要素）に対応して、上記アドレス及びデータ型等を含む属性定義ファイル606の1要素分のデータ

(7)

12

が作成される（ステップ707）。そして、その作成された1要素分が、属性定義ファイル606に書き込まれる（ステップ708）。

【0066】最後に、ステップ705で修正された現在行の内容が、画面定義ファイル605に書き込まれる（ステップ709）。以上の第3の実施の形態により、画面定義ファイル及び属性定義ファイルを、本体装置にロードされるテーブルに対応するロードモジュールファイルと、その元となるプログラムソースファイルから自動生成することができるため、システムの開発者は、ほとんど作業を必要とせずにテーブル変更部の機能を提供することができ、テーブル変更プログラム301の開発効率を向上させることができる。

【0067】また、本体装置のテーブル構造と保守端末のテーブル変更部の機能の間で、テーブル構造の認識違いに基づくバグの発生がなくなるため、テーブル変更部の機能の信頼性を向上させることができる。

第4の実施の形態

図8は、本発明の第4の実施の形態の構成図である。この構成は、図1の本発明の第1の実施の形態とほとんど同様である。

【0068】第4の実施の形態では、第1の実施の形態において、テーブル変更部105が実現するテーブル変更機能にリードオンリーモードが設けられ、テーブル受信部103の動作とテーブル変更部105の動作がバッチ処理化されて一連の動作として実行されることにより、本体装置のテーブルを単に表示する機能及び装置状態を表示する機能を簡単に付加することができる。

【0069】本体装置113内のテーブル領域111のテーブル内容を表示するための上記バッチ処理が起動されると、まず、保守端末112のテーブル受信部103が起動されることにより、本体装置113内のテーブル領域111から保守端末112にテーブル内容が転送される。具体的には、本体装置113内のテーブル管理部108が、メモリ獲得部110を起動することによって、テーブル領域111のテーブル内容が読み出される。このテーブル内容は、本体装置113内の本体との通信部106と保守端末112内の本体との通信部106を介して、テーブル受信部103に転送される。テーブル受信部103は、そのテーブル内容を、テーブルファイル部101に一時テーブルファイルとして格納する。続いて、テーブル変更部105が起動されることにより、テーブルファイル部101内の上記一時テーブルファイルの内容であるテーブル内容がリードオンリーモードで表示される。

【0070】一方、本体装置113内の装置状態領域801の内容を表示するための上記バッチ処理が起動されると、まず、保守端末112のテーブル受信部103が起動されることにより、本体装置113内の装置状態領域801から保守端末112に装置状態が転送される。

具体的には、本体装置113内のテーブル管理部108が、メモリ獲得部110を起動することによって、装置状態領域801の装置状態が読み出される。この装置状態は、本体装置113内の本体との通信部106と保守端末112内の本体との通信部106を介して、テーブル受信部103に転送される。テーブル受信部103は、その装置状態を、テーブルファイル部101にやはり一時テーブルファイルとして格納する。なお、テーブル内容も装置状態もメモリ領域に配置されているデータであるため、装置状態も一時テーブルファイルとして格納することが可能である。続いて、テーブル変更部105が起動されることにより、テーブルファイル部101内の上記一時テーブルファイルの内容である装置状態がリードオンリーモードで表示される。

【0071】以上の第4の実施の形態により、第1の実施の形態の構成におけるテーブル変更部105の機能にリードオンリーモードが付加するだけで、本体装置113のテーブル表示機能及び装置状態表示機能を簡単に実現できる。

第5の実施の形態

図9は、本発明の第5の実施の形態の構成図である。この構成は、図1の本発明の第1の実施の形態の構成に基づいている。

【0072】第5の実施の形態では、保守端末112による本体装置113内のテーブル領域111内のテーブル内容の変更履歴を管理できることが特徴である。第1の実施の形態の方式に基づいて保守端末112上で変更されたテーブルファイル部101内の新テーブルファイルの内容が保守端末112内のテーブル送信部102により本体装置113に転送されると、その新テーブルファイルの内容は、まず、本体装置113内のテーブル管理部108によって受信される。

【0073】テーブル管理部108は、受信した新テーブルファイルの内容を、テーブル変更履歴管理部902に受け渡す。テーブル変更履歴管理部902は、まず、テーブル領域111の旧テーブル内容を読み取り、テーブル管理部108から受け取った新テーブルファイルの内容と上記旧テーブル内容とを比較して、両者で異なる部分を本体装置113内の変更履歴領域902に格納し、その後に、メモリ格納部109を介して、テーブル領域111に新テーブルファイルの内容を格納する。

【0074】一方、保守端末112を操作するオペレータからテーブル変更履歴の表示要求が発生すると、テーブル変更履歴表示部901が起動される。テーブル変更履歴表示部901は、まず、本体装置113の変更履歴領域902内のテーブル変更履歴を取得する。具体的には、本体装置113内のテーブル変更履歴管理部902が変更履歴領域902内のテーブル変更履歴を読み出し、それをテーブル管理部108へ受け渡す。テーブル管理部108は、上記テーブル変更履歴を、本体装置1

13内の保守端末との通信部107及び保守端末112内の本体との通信部106を介して、テーブル変更履歴表示部901へ転送する。

【0075】テーブル変更履歴表示部901は、本体装置113から受信したテーブル変更履歴を、オペレータの見やすい形式に変換して、特には図示しない表示装置に表示する。

【0076】以上第5の実施の形態により、本体装置113がテーブル領域111のテーブル内容の変更日時及び変更内容等を自動的に管理するため、テーブル版数管理の安全性を向上させることが可能となる。

第6の実施の形態

図10は、本発明の第6の実施の形態の構成図である。

【0077】第6の実施の形態では、テーブル変更を実施している間に本体装置を停止させる必要がないことが特徴である。まず、図11に示されるように、本体装置1007の運用状態において、第1の実施の形態の場合と同様にしてテーブルファイルが保守端末1006側で変更され、保守端末1006内のテーブルファイル部1001内の新テーブルファイルが本体装置1007に転送される。この新テーブルファイルの内容は、本体装置1007内の第1テーブル領域1002に格納される。この間、本体装置1007内の各アプリケーション1005は、第2テーブル領域1004を参照しながら運用を続行している。

【0078】次に、適当なタイミングで、本体装置1007が再起動され、図11に示されるように、この再起動の間のみ本体装置1007は運用停止状態となる。上記再起動のシーケンスにおいて、本体装置1007内のテーブル変換部1003は、第1テーブル領域1002に新テーブルファイルの内容が書き込まれている場合には、その内容を各アプリケーション1005が参照しやすい形式に変換し、その変換後のテーブル内容を第2テーブル領域1004に書き込む。

【0079】上記再起動シーケンスの後、本体装置1007が運用を再開すると、各アプリケーション1005は、第2テーブル領域1004に書き込まれた新たなテーブル内容に従って運用を再開する（図11参照）。

【0080】以上第6の実施の形態により、本体装置1007内の各アプリケーション1005は、変換前の第1テーブル領域1002のテーブル内容を参照することができないため、本体装置1007が運用状態にあるときに保守端末1006から本体装置1007に新テーブルファイルを転送することが可能となる。このため、オペレータは、テーブルの変更時には、保守端末1006から本体装置1007へ新テーブルファイルを転送した後、本体装置1007を再起動（リセット）するだけでもなく、テーブル変更時の操作を大幅に簡略化することができる。

【0081】また、装置の運用停止時間を短縮すること

ができる、装置稼働率を向上させることができる。また、変換後のテーブル内容は、各アプリケーション1005が参照しやすい形式を有するため、各アプリケーション1005のためのプロセッサの実行時間を短縮することができる。

【0082】更に、システムの開発者は、本体装置1007側に、メンテナンスモード等の保守を行うモードを実装する必要がなくなり、開発効率を向上させることができとなる。

第7の実施の形態

図12は、本発明の第7の実施の形態の構成図である。

【0083】第7の実施の形態では、マルチプロセッサシステムにおいて、全てのプロセッサで共通的に使用するテーブルは、共有メモリに配置されないことが特徴である。

【0084】図12において、まず、システム管理CPU1212内のテーブル変換部1203は、第6の実施の形態の場合と同様にして、リセット時に保守端末1211から第1テーブル領域1202に受信されている新テーブル内容を、アプリケーション1205が参照しやすい形式に変換し、その変換後のテーブル内容を第2テーブル領域1204に書き込む。

【0085】更に、システム管理CPU1212内のテーブル分配部1206は、CPU間通信機能を使用して、第2テーブル領域1204に書き込まれた新テーブル内容を、個別機能CPU1213に送信する。

【0086】各個別機能CPU1213内のテーブル受信部1207は、システム管理CPU1212から新テーブル内容を受信すると、それを自装置内のローカルテーブル領域1208に格納し、自装置の運用を開始する。

【0087】システム管理CPU1212内の各アプリケーション1205は、第2テーブル領域1204に書き込まれた新たなテーブル内容に従って運用を開始する。また、個別機能CPU1213内の各アプリケーション1209は、自装置内のローカルテーブル領域1208に書き込まれた新たなテーブル内容に従って運用を開始する。

【0088】以上の第7の実施の形態により、テーブル内容が各プロセッサのローカルテーブル領域1208(ローカルメモリ上)に配置されるため、該テーブルをアクセスするプロセッサの実行時間を短縮することができる。

【0089】また、共有メモリ1210へのアクセスの集中を回避することができ、システム全体のレスポンスを向上させることができる。

【0090】

【実施例】図13は、第1～第7の実施の形態を実現するシステムの実施例の構成図である。この実施例は、共有メモリを有する密結合マルチプロセッサ装置を使用し

たHDL C(ハイレベルデータリンク手順)回線制御機能と、個別出力機能を有するオンラインリアルタイムシステムに、本発明を適用したものである。

【0091】このシステムは、システム管理CPU1320、HDL C回線制御CPU1321、及び個別出力制御CPU1322から構成される。図13に示されるシステム(以下、本体装置ともいう)上で稼働するソフトウェアは、図14に示される構成を有するソフト開発用パソコン1408上で開発されるものとする。

【0092】また、図13に示されるシステム(本体装置)は、図15に示される保守端末1518と接続する機能を有する。図14に示されるソフト開発用パソコン1408において、図13に示される本体装置上で稼働するソフトウェアが開発されるが、そのソフトウェアのプログラムソースは、大きく分けて本体装置上で実行されるプログラムのソースファイル1401と、本体装置上に置かれるテーブルソースファイル1402とからなる。

【0093】これらのファイルがコンパイル、リンク、及びロケートされると、本体装置ロードモジュールファイル1403とロケートマップファイル1404とが生成される。本体装置ロードモジュールファイル1403は、本体装置にロードして使用されるファイルである。ロケートマップファイル1404は、本体装置で実行される各プログラム、テーブル内のシンボルのロケートアドレス等を含むファイルである。

【0094】ソフト開発用パソコン1408内のテーブル変更用ファイル自動生成プログラム1405は、上記テーブルソースファイル1402及びロケートマップファイル1404に基づいて、画面定義ファイル1406及び属性定義ファイル1407を生成する。

【0095】図16に、テーブルソースファイル1402の実施例を示す。この実施例では、本ファイルは、C言語によって記述されている。図16から、図13に示される本体装置は、システム定義テーブル、HDL C回線定義テーブル、及び個別出力定義テーブルを保有することがわかる。

【0096】図17に、ロケートマップファイル1404の実施例を示す。本ファイルは、図15に示されるテーブルソースファイル1402内の各テーブル要素のシンボル名のロケートアドレスを含んでいる。

【0097】図18に、画面定義ファイル1406の実施例を示す。本ファイルは、図15に示されるテーブルソースファイル1402及び図16に示されるロケートマップファイル1404から自動生成されたファイルである。本ファイル中の“\$\$”は、テーブル変更時に入力対象となる項目を示している。

【0098】図19に、属性定義ファイル1407の実施例を示す。本ファイルは、図15に示されるテーブルソースファイル1402及び図16に示されるロケート

マップファイル1404から自動生成されたファイルである。図18に示される画面定義ファイル1406中の各入力項目の属性として、本体装置メモアドレス、入力形式、表示色、及びデータ型を有する。

【0099】次に、図15に示される保守端末1518においては、図14に示されるソフト開発用パソコン1408からコピーされた画面定義ファイル1501（図14の画面定義ファイル1406に対応する）及び属性定義ファイル1502（図14の属性定義ファイル1407に対応する）に基づいて、テーブル変更プログラム1503が動作することにより、テーブル変更機能が実現され、テーブルファイル部1504にテーブルファイルが生成される。

【0100】図20に、このテーブル変更機能によって実現されるテーブル変更画面の実施例を示す。反転項目が、入力項目である。このようにして、本体装置のプログラムファイルから自動的にテーブル変更機能を提供することができる。

【0101】上述のテーブル変更機能によりテーブルファイル部1504内に生成されたテーブルファイルは、テーブル送信部1505、テーブル受信部1506、及びテーブル比較部1507により、図13に示される本体装置との間のテーブルファイルの送信、受信、比較動作が実現される。

【0102】また、ソフト開発用パソコン1408からコピーされた画面定義ファイル1501及び属性定義ファイル1502に基づいて、テーブル変更プログラム1503がリードオンリーモードで動作し、そのテーブル変更プログラム1503とテーブル受信部1510の受信機能と一時テーブルファイル部1509とが連携して動作することで、本体装置内テーブルの表示機能が実現される。

【0103】更に、画面定義ファイル1511及び属性定義ファイル1512に基づいて、テーブル変更プログラム1513がリードオンリーモードで動作し、そのテーブル変更プログラム1513とテーブル受信部1511の受信機能と一時テーブルファイル部1514とが連携して動作することで、本体装置内装置状態の表示機能が実現される。

【0104】一方、保守端末1518内に、テーブル変更履歴表示部1516が設けられることにより、本体装置内のテーブルの変更日時、変更箇所を表示することが可能となる。

【0105】図21に、図13に示される本体装置内のテーブル変更履歴領域1308の実施例を示す。この領域には、複数個の変更履歴を格納することができる。また、変更履歴領域が一杯になった時には最新の変更履歴が最も古い履歴に上書きされることで、この領域には常に最新のn個の履歴が格納される。各変更履歴は、変更日時、変更アドレス数、及び（変更アドレス、旧デー

タ、新データの組）とから構成される。

【0106】図22に、テーブル変更履歴表示画面の実施例を示す。保守端末1518内のテーブル変更履歴表示部1516は、図13に示される本体装置内のテーブル変更履歴領域1308内のテーブル変更履歴を受け取り、その内容を翻訳して表示する。図22に示されるように、変更日時と、変更箇所のみが表示される。

【0107】図13に示される本体装置では、テーブル管理部1302が、保守端末との通信部1301を介して図15に示される保守端末1518から受信したテーブルファイルを、テーブル変更履歴管理部1303及びメモリ格納部1304を介して第1テーブル領域1306に格納する。また、テーブル管理部108は、装置状態領域1307からメモリ獲得部1305及びテーブル変更履歴管理部1303を介して獲得した装置状態を、保守端末との通信部1301を介して保守端末1518に送信する。

【0108】テーブル変更履歴管理部1303は、保守端末1518から受信された新テーブルファイルの内容が第1テーブル領域1306に書き込まれる際に、第1テーブル領域1306からメモリ獲得部1305を介して旧テーブル内容を獲得し、新テーブルファイルの内容と旧テーブル内容とを比較することにより、テーブル変更履歴を自動的にテーブル変更履歴領域1308に保存する。また、テーブル変更履歴管理部1303は、保守端末1518からの要求時に、テーブル変更履歴領域1308に保存されているテーブル変更履歴を、テーブル管理部1302及び保守端末との通信部1301を介して保守端末1518に送信する。

【0109】システム管理CPU1320内のテーブル変換部1309は、リセット時に、第1テーブル領域1306内の新テーブル内容を、システム管理部1311、HDL C回線制御部1315、及び個別出力部1318等の各アプリケーションが参照しやすい形式に変換し、その変換されたテーブル内容を第2テーブル領域1310に転送する。

【0110】更に、システム管理CPU1320内のテーブル分配部1312は、共有メモリ1319経由のメールボックスを使用して、HDL C回線制御CPU1321にHDL C回線定義テーブルの内容を、また、個別出力制御CPU1322に個別出力定義テーブルの内容を、それぞれ送信する。

【0111】HDL C回線制御CPU1321内のテーブル受信部1313は、システム管理CPU1320からテーブル内容を受信すると、それを自装置内のローカルメモリ上のHDL Cテーブル領域1314に格納し、自装置の運用を開始する。

【0112】個別出力制御CPU1322内のテーブル受信部1316は、システム管理CPU1320からテーブル内容を受信すると、それを自装置内のローカルメ

モリ上の個別出力テーブル領域1317に格納し、自装置の運用を開始する。

【0113】システム管理CPU1320のシステム管理部1311は、第2テーブル領域1310に書き込まれた新たなテーブル内容に従って、システム管理の運用を開始する。また、HCLC回線制御システム1321内のHDLG回線制御部1315は、自装置内のHDLGテーブル領域1314に書き込まれた新たなテーブル内容に従って、HDLG回線制御の運用を開始する。同様に、個別出力制御CPU1322内の個別出力部1318は、自装置内の個別出力テーブル領域1317に書き込まれた新たなテーブル内容に従って、個別出力動作の運用を開始する。

【0114】

【発明の効果】本発明の第1の態様によれば、テーブル内容が保守端末上のテーブルファイルとして管理されることにより、本体装置がない場所（事務所等）でのテーブル変更作業等が可能となり、オペレータは事前にテーブル内容を準備することが可能となり、テーブル変更の安全性が向上する。

【0115】次に、テーブル変更機能に定義ファイルを用いる構成により、システム開発者は、画面イメージとテーブル構造の定義ファイルを作成するだけで、テーブル変更機能を簡単に提供することが可能となり、開発効率が向上する。

【0116】次に、定義ファイル生成手段を有する構成により、システム開発者はほとんど作業なしでテーブル変更機能を提供でき、開発効率を向上させることができる。また、本体装置のテーブル構造と保守端末のテーブル変更手段の機能の間で、テーブル構造の認識違いに基づくバグの発生がなくなるため、テーブル変更手段の機能の信頼性を向上させることができる。

【0117】次に、テーブル受信手段とリードオンリーモードを有するテーブル変更手段を連携動作させる構成により、本体装置のテーブル内容や装置状態の表示機能を簡単に実現することができる。

【0118】次に、更新履歴を管理可能な構成により、テーブル版数管理の安全性を向上させることができる。次に、テーブル変換手段を含む構成により、変更されたテーブル内容が保守端末から本体装置に転送される際に本体装置の運用を停止させる必要がなく、転送終了後に本体装置をリセットするだけで本体装置のテーブル内容を更新することができるため、本体装置にメンテナンスマード等の保守を行うモードを実装する必要がなく、また、本体装置の稼働率の低下も防ぐことができる。

【0119】また、本発明の第2の態様によれば、テーブル内容が各プロセッサにローカルに保持されるため、テーブル内容をアクセスするプロセッサの実行時間を短縮することができる。また、共有メモリへのアクセスの集中を回避することができ、システム全体のレスポンス

を向上させることが可能となる。

【図面の簡単な説明】

【図1】第1の実施の形態の構成図である。

【図2】第1の実施の形態の動作フローチャートである。

【図3】第2の実施の形態の構成図である。

【図4】第2の実施の形態におけるテーブル変更プログラムの動作フローチャートである。

【図5】図4のステップ404の詳細を示す動作フローチャートである。

【図6】第3の実施の形態の構成図である。

【図7】第3の実施の形態におけるテーブル変更用ファイル自動生成プログラムの動作フローチャートである。

【図8】第4の実施の形態の構成図である。

【図9】第5の実施の形態の構成図である。

【図10】第6の実施の形態の構成図である。

【図11】第6の実施の形態におけるテーブル変更手順を示す図である。

【図12】第7の実施の形態の構成図である。

【図13】システムの実施例の構成図である。

【図14】ソフト開発用パソコンの実施例の構成図である。

【図15】保守端末の実施例の構成図である。

【図16】テーブルソースファイルの実施例を示す図である。

【図17】ロケートマップファイルの実施例を示す図である。

【図18】画面定義ファイルの実施例を示す図である。

【図19】属性定義ファイルの実施例を示す図である。

【図20】テーブル変更画面の実施例を示す図である。

【図21】テーブル変更履歴領域の実施例を示す図である。

【図22】テーブル変更履歴表示画面の実施例を示す図である。

【図23】従来技術の説明図である。

【図24】従来技術でのテーブル変更手順を示す図である。

【図25】マルチプロセッサシステムでの従来のテーブル配置技術の説明図である。

【符号の説明】

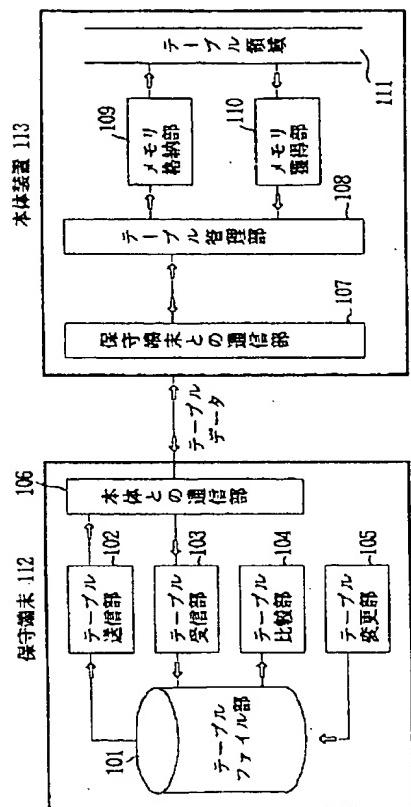
101	テーブルファイル
102	テーブル送信部
103	テーブル受信部
104	テーブル比較部
105	テーブル変更部
106	本体との通信部
107	保守端末との通信部
108	テーブル管理部
109	メモリ格納部
110	メモリ獲得部

21

1 1 1 テーブル領域
1 1 2 保守端末

【図1】

第1の実施の形態の構成図

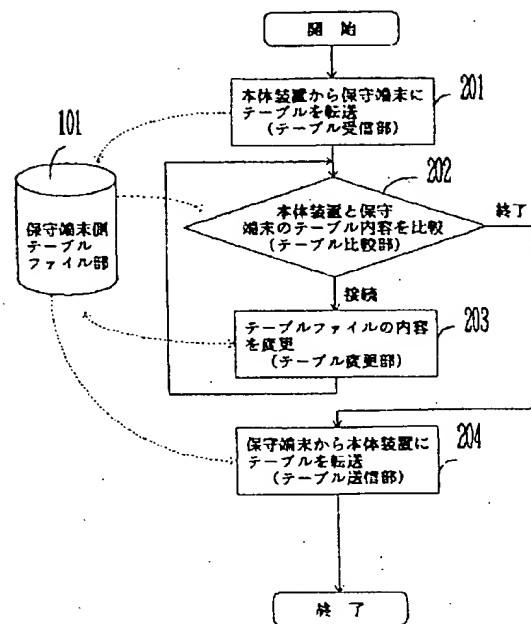


22

1 1 3 本体装置

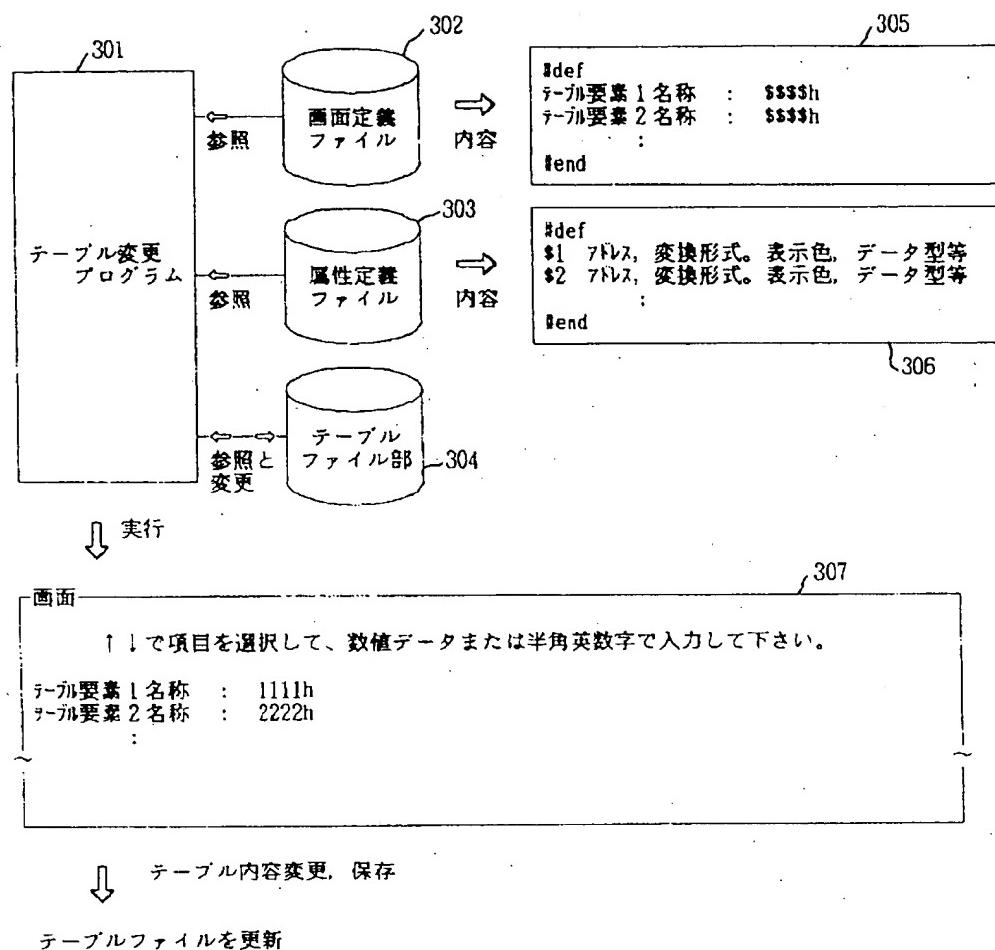
【図2】

第1の実施の形態の動作フロー チャート



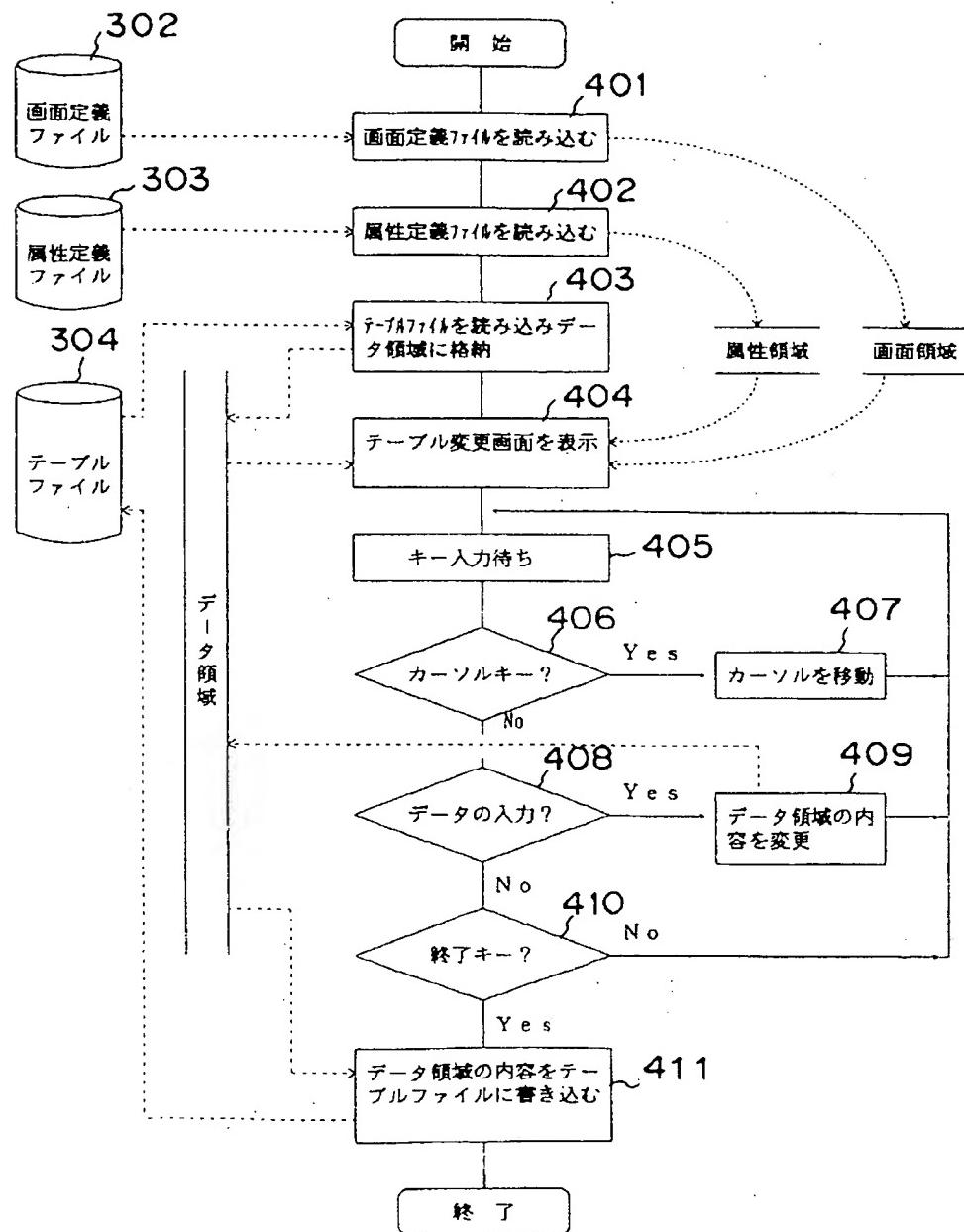
[図3]

第3の実施の形態の構成図



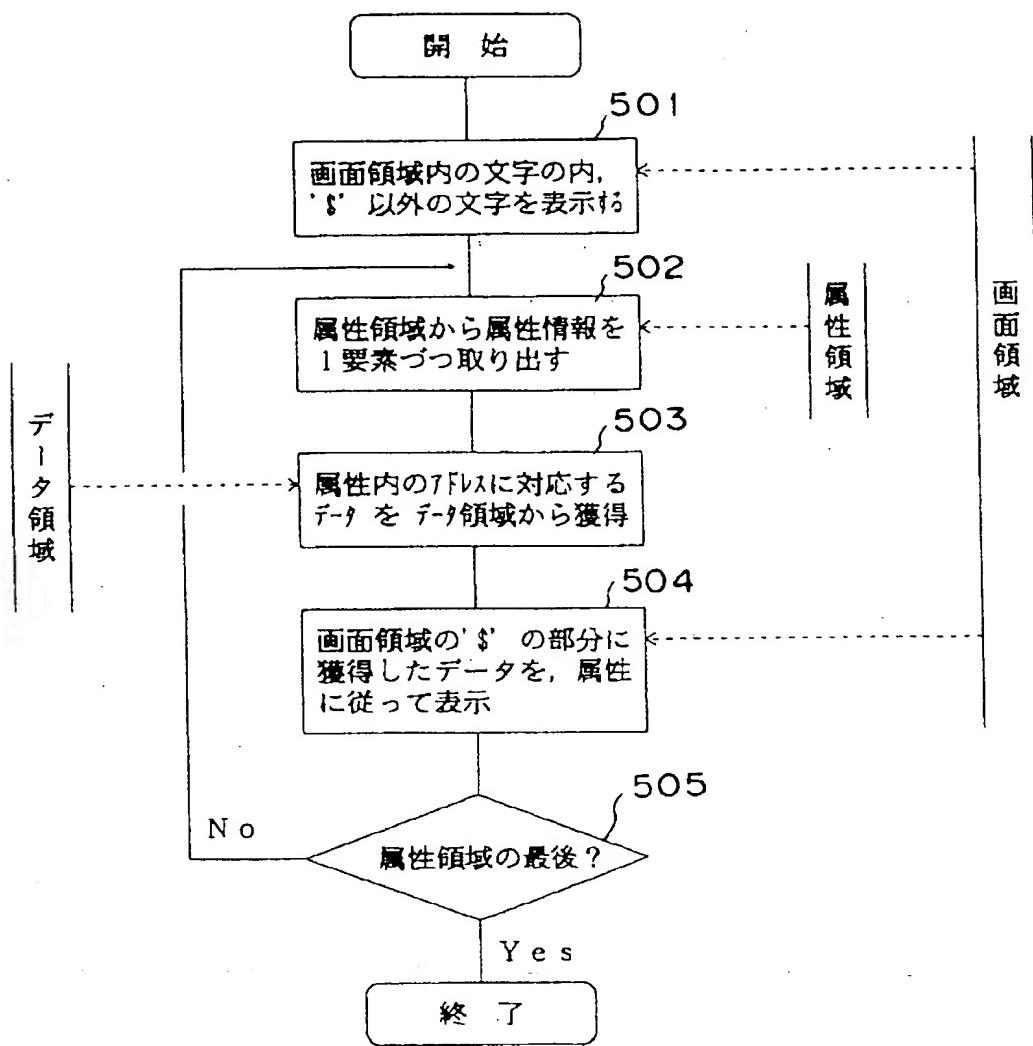
【図4】

第2の実施の形態における
テーブル変更プログラムの動作フローチャート



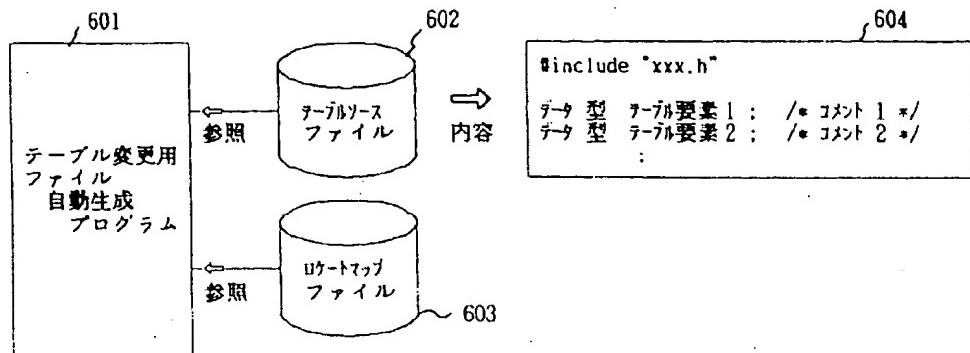
【図5】

図4のステップ404の
詳細を示す動作フローチャート



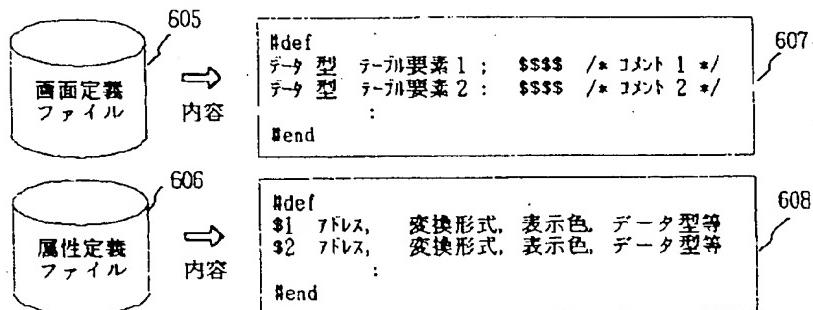
【図6】

第3の実施の形態の構成図



↓ 実行

下記2ファイルを生成

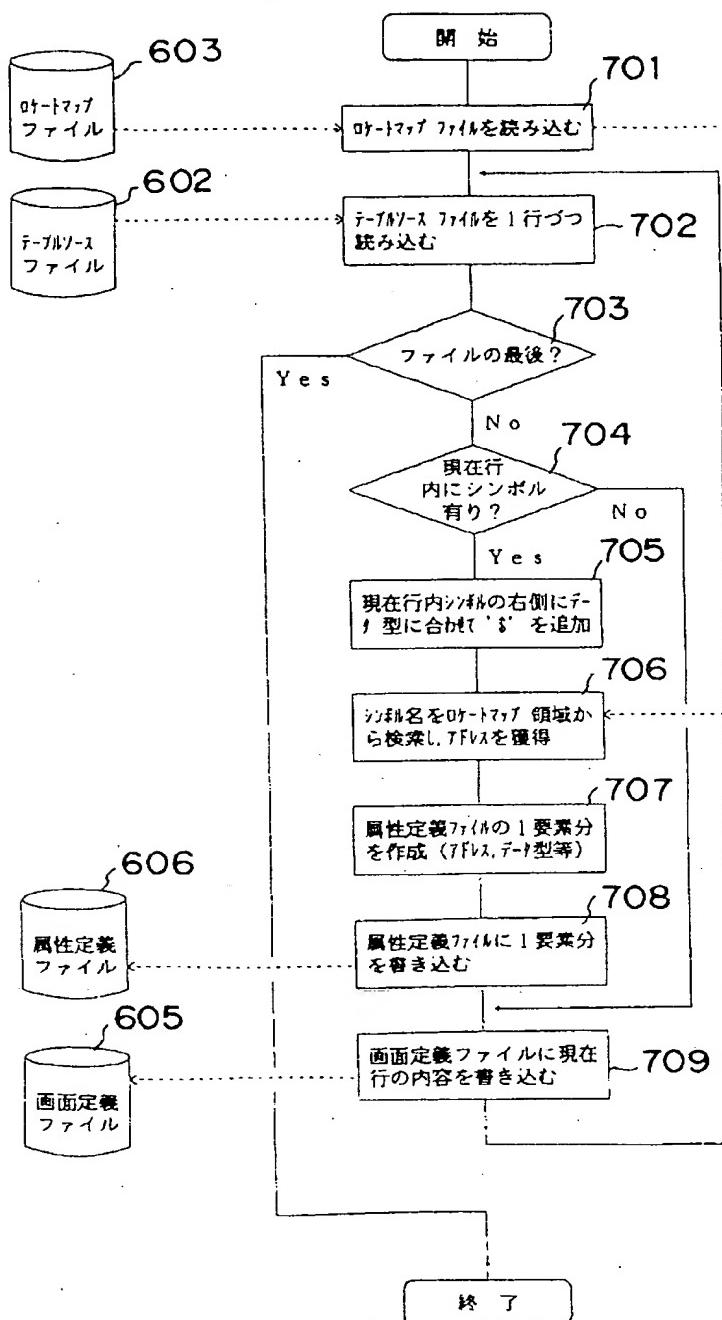


↓

テーブル変更プログラムで使用

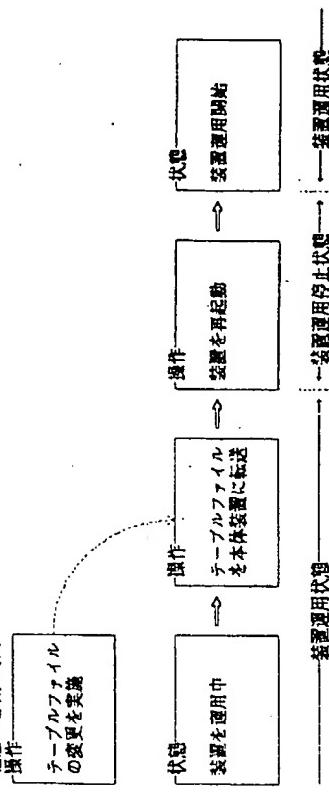
[図7]

第3の実施の形態における テーブル変更用ファイル自動生成プログラムの 動作フロー-チャート



[図11]

第6の実施の形態における テーブル変更手順を示す図



[図 8]

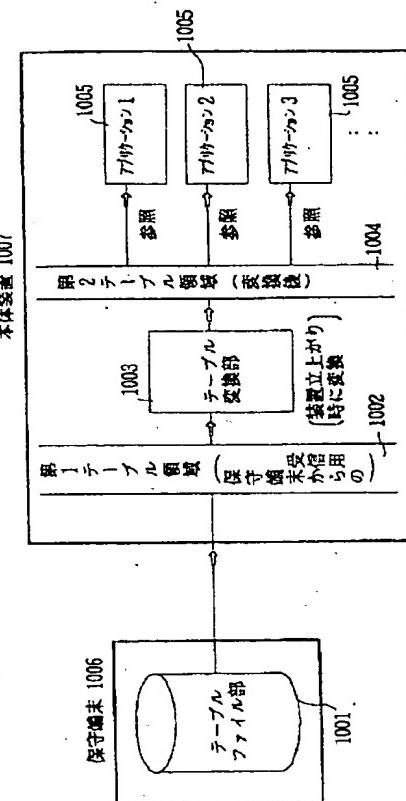
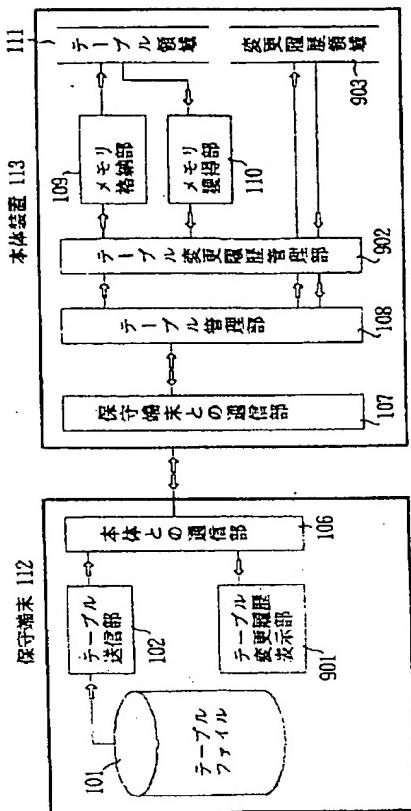
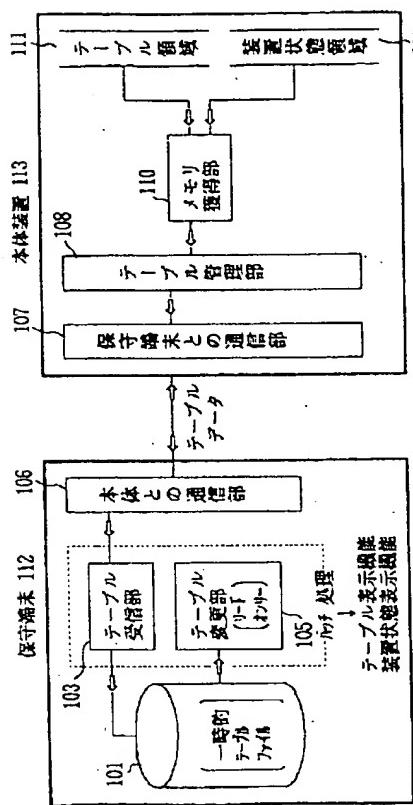
[図9]

【四 10】

第4の実施の形態の構成・図

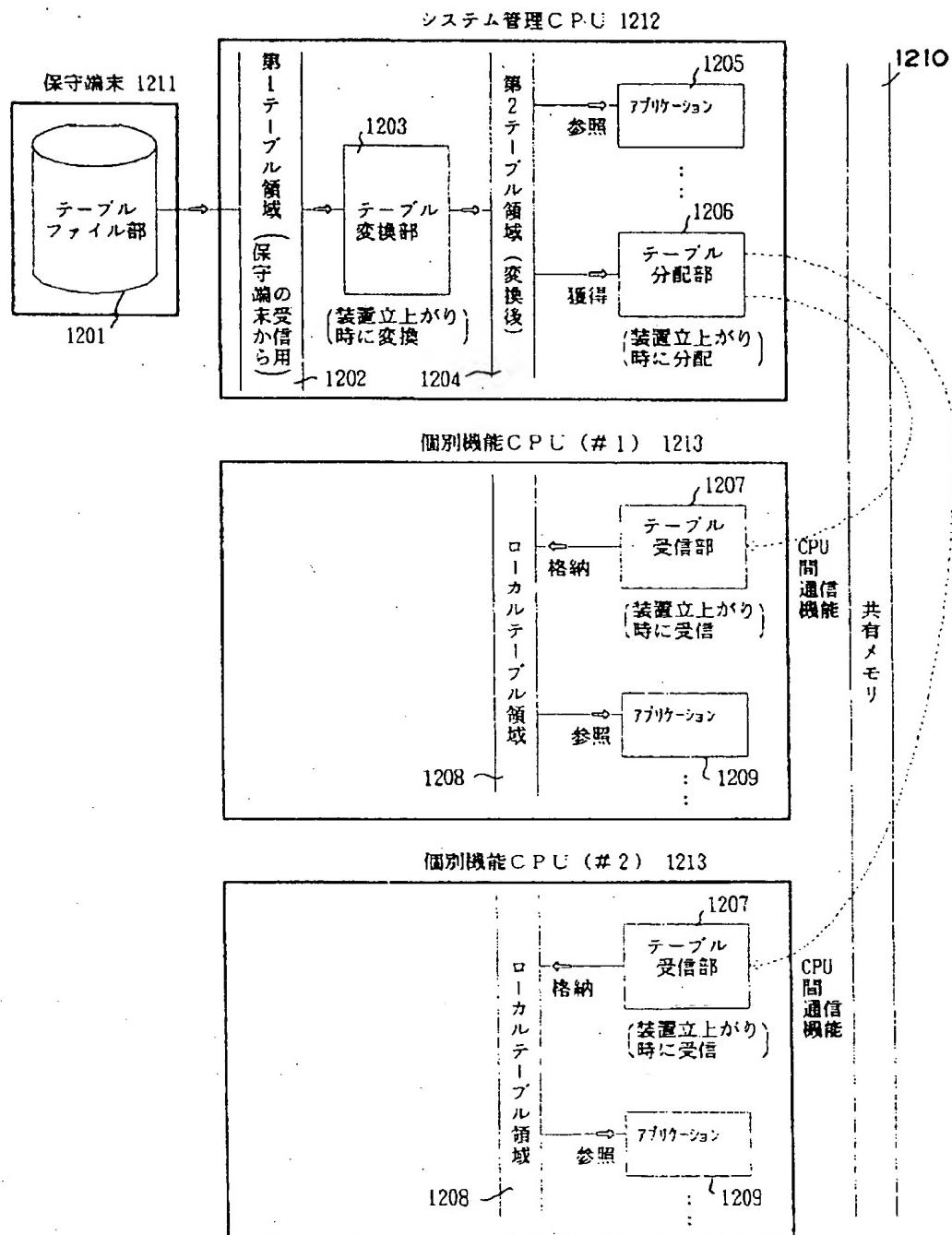
第5の実施の形態の構成図

第6の実施の形態の構成図



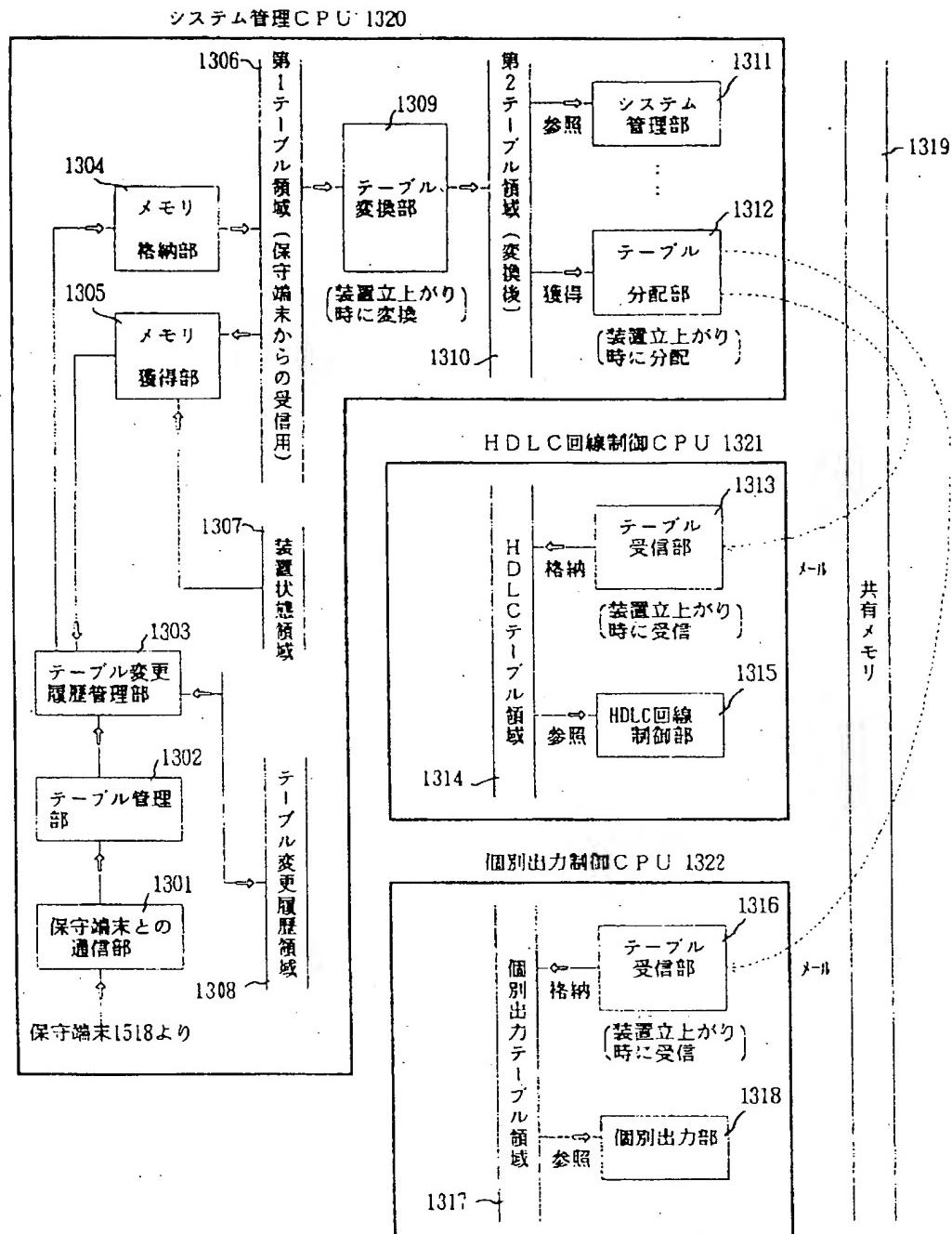
【図12】

第7の実施の形態の構成図



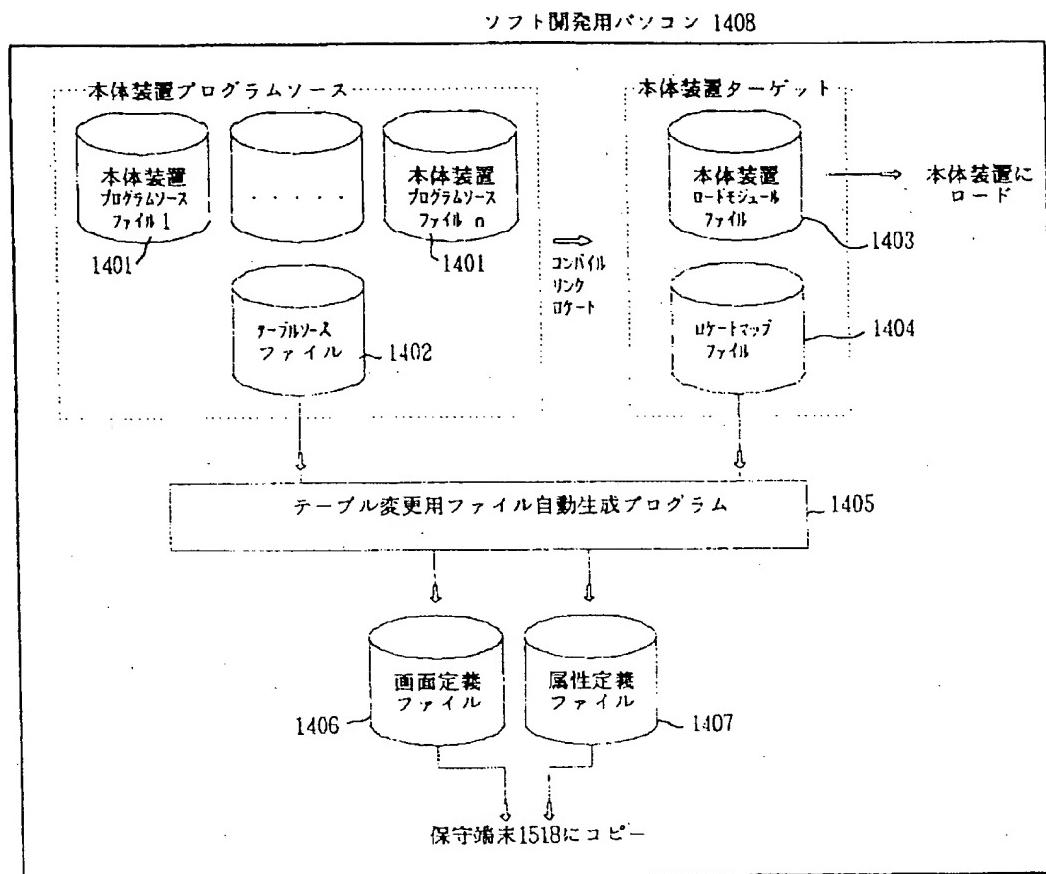
[図13]

シス テ ム の 実 施 例 の 構 成 図



【図14】

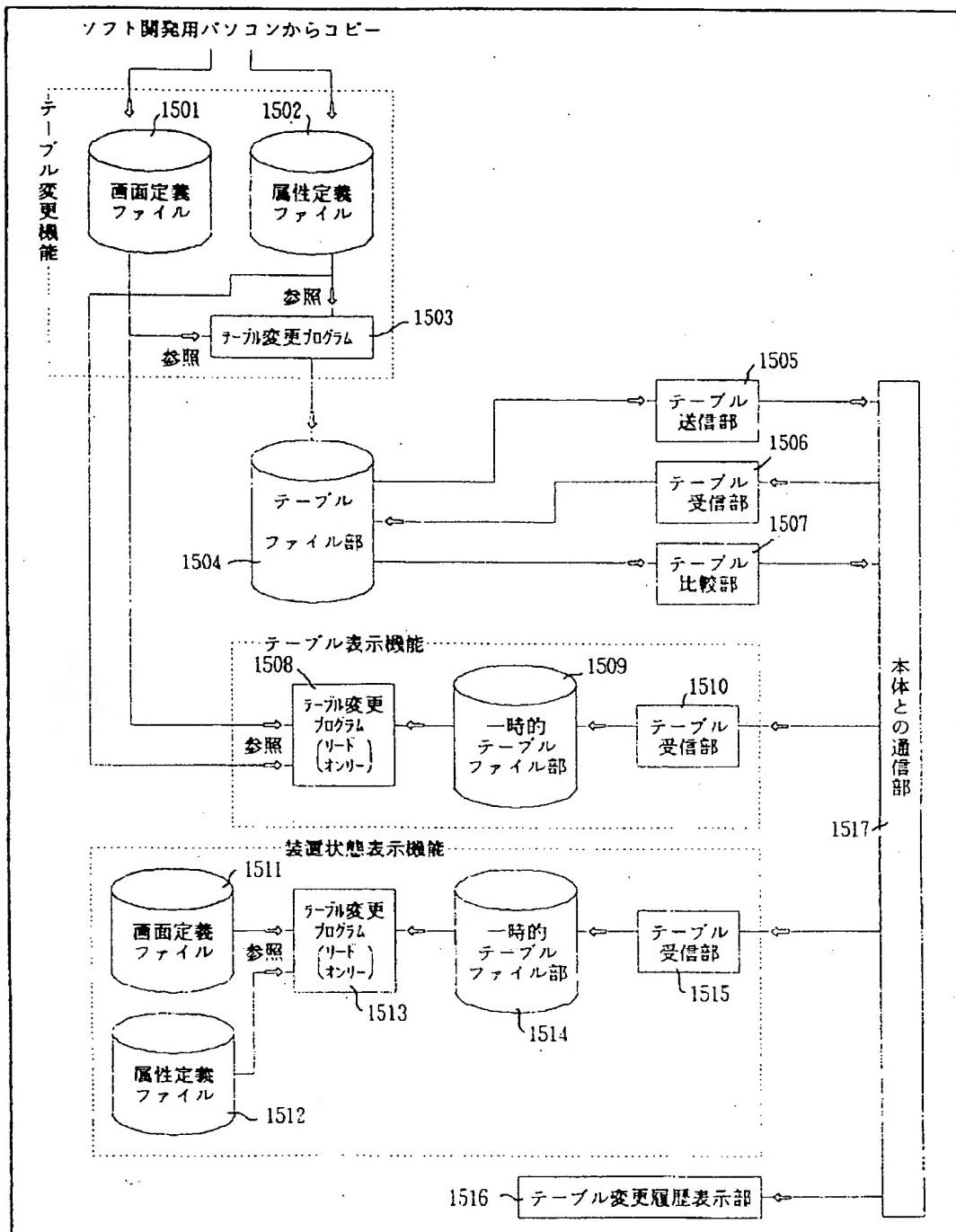
ソフト開発用パソコンの実施例の構成図



【図15】

保守端末の実施例の構成図

保守端末 1518



【図16】

【図17】

【図18】

テーブルソースファイルの実施例を示す図 ロケートマップファイルの実施例を示す図 画面定義ファイルの実施例を示す図

```

/* 235定義-71 */
char spcpu; /* 汎用CPU 実装状態 (1:実装, 0:未実装) */
char holccpu; /* HOLC回線CPU 実装状態 (1:実装, 0:未実装) */
char pocpu; /* 個別出力CPU 実装状態 (1:実装, 0:未実装) */

/* HOLC回線定義-71 */
char hijacu; /* HOLC回線 実装状態 (1:実装, 0:未実装) */
unsigned pload; /* 回路速度 (4000/5600/19200/48000) */
char hijadr; /* 自局アドレス */

/* 個別出力定義-71 */
char pijscpu; /* 個別出力 実装状態 (1:実装, 0:未実装) */
char hijadr; /* 回路速度 (4000/5600/19200/48000) */
char pload; /* 出力種別 (0:通常出力, 1:IF出力) */
char pkind; /* 自局アドレス */

```

SYMBOL TABLE					
BASE	OFFSET	TYPE	SYMBOL	BASE	OFFSET
1500h	0000h	PUB	syscpu	1500h	0001h
1500h	0002h	PUB	pocpu		
1500h	0100h	PUB	hijscpu	1500h	0101h
1500h	0103h	PUB	hijadr		
1500h	0200h	PUB	pjisscpu	1500h	0201h
1500h	0203h	PUB	pkind		

```

/* 235定義-71 */
char syscpu; /* 汎用CPU 実装状態 (1:実装, 0:未実装) */
char holccpu; /* HOLC回線CPU 実装状態 (1:実装, 0:未実装) */
char pocpu; /* 個別出力CPU 実装状態 (1:実装, 0:未実装) */

/* HOLC回線定義-71 */
char hijacu; /* HOLC回線 実装状態 (1:実装, 0:未実装) */
unsigned pload; /* 回路速度 (4000/5600/19200/48000) */
char hijadr; /* 自局アドレス */

/* 個別出力定義-71 */
char pijscpu; /* 個別出力 実装状態 (1:実装, 0:未実装) */
char hijadr; /* 回路速度 (4000/5600/19200/48000) */
char pload; /* 出力種別 (0:通常出力, 1:IF出力) */
char pkind; /* 自局アドレス */

/* 個別出力定義-71 */
char hijacu; /* 個別出力 実装状態 (1:実装, 0:未実装) */
unsigned pload; /* 回路速度 (4000/5600/19200/48000) */
char hijadr; /* 出力種別 (0:通常出力, 1:IF出力) */
char pkind; /* 自局アドレス */

/* 入力項目 */

```

【図19】

属性定義ファイルの実施例を示す図

```

#define
    本体装置番号
    入力表示 (x:位置, y:位置, z:位置)
    示色 (w:白, y:黄)
    フォント (b:[1]font, w:[1]font)

    //HUC回路CPU 実装状態
    //HUC回路CPU
    //個別出力CPU
    //HUC回路 実装状態
    //回路速度
    //自局バス

    //個別出力 実装状態
    //I/Oバス
    //出力種別

$001 1500. 0000. d, w, b <
$002 1500. 0001. d, w, b <
$003 1500. 0002. d, w, b <
$011 1500. 0100. d, w, b <
$012 1500. 0101. d, w, b <
$013 1500. 0103. d, w, b <
$021 1500. 0200. d, w, b <
$022 1500. 0201. d, w, b <
$023 1500. 0203. d, w, b <

tend

```

【図20】

テーブル変更画面の実施例を示す図

↑で項目を選択して、該当データまたは半角英数字で入力して下さい。

```

/* 仕様定義 */
    /* HUC回路CPU 実装状態 (1:実装, 0:未実装) */
    char syncpu; /* HUC回路CPU 実装状態 (1:実装, 0:未実装) */
    char hidcpu; /* 個別出力CPU 実装状態 (1:実装, 0:未実装) */
    char pcpu; /* 自局バス */

/* HUC回路定義 */
    /* HUC回路 ; */ /* HUC回路 (4800/9600/19200/24000/48000) */
    unsigned baudr; /* 個別出力 ; */ /* 個別出力 (0:通常出力, 1:I/O付) */
    char baddr; /* I/Oバス */

/* 個別出力定義 */
    /* 個別出力 ; */ /* 個別出力 (1:実装, 0:未実装) */
    unsigned pladdr; /* I/Oバス */ /* I/Oバス (0:未実装, 1:実装) */
    char plkind; /* 出力種別 */ /* 出力種別 (0:通常出力, 1:I/O付) */

```

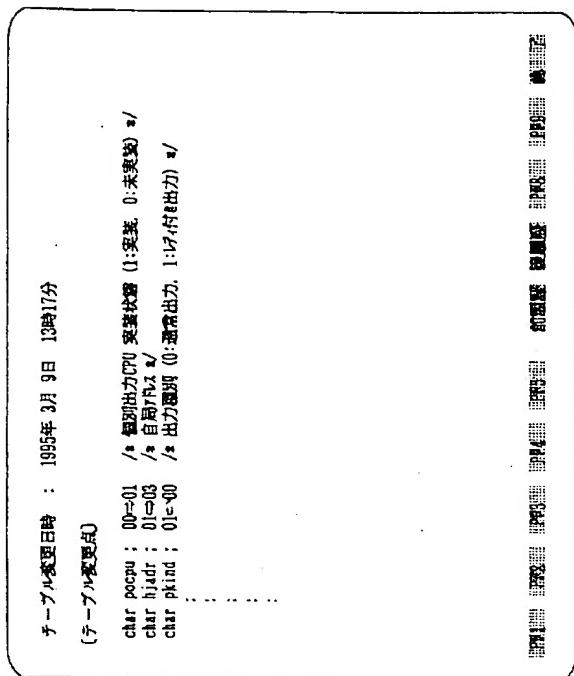
【図21】

テーブル変更履歴領域の実施例を示す図

変更日時	
変更アドレス数	
変更履歴1	"
変更履歴2	"
変更履歴3	"
変更履歴4	"
変更履歴5	"
変更履歴6	"
変更履歴7	"
変更履歴8	"
変更履歴9	"
変更履歴10	"

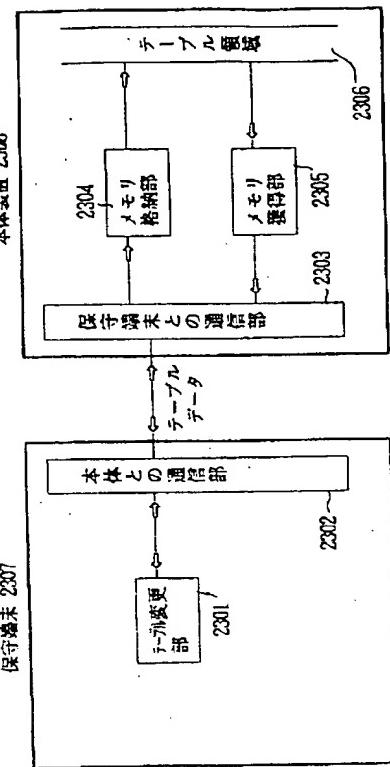
【図22】

テーブル変更履歴表示画面の実施例を示す図



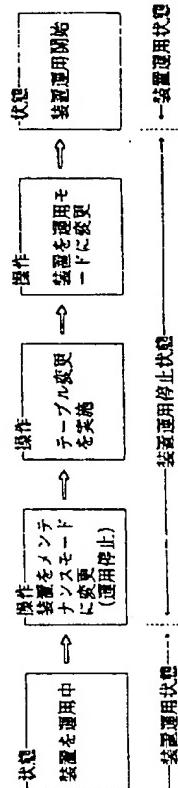
【図23】

従来技術の説明図



【図24】

従来技術でのテーブル変更手順を示す図



【図25】

マルチプロセッサシステムにおける
従来のテーブル配置技術の説明図

